

Universidade Federal do Rio de Janeiro

Núcleo de Computação Eletrônica

Carlos Arthur Rodrigues Lima

GERENCIAMENTO EM GRID

Rio de Janeiro

2006

Carlos Arthur Rodrigues Lima

## GERENCIAMENTO EM GRID

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ

Prof. Luci Pirmez – Orientadora  
D.Sc., COPPE/UFRJ, Brasil

Rio de Janeiro

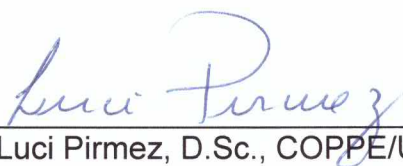
2006

Carlos Arthur Rodrigues Lima

## GERENCIAMENTO EM GRID

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ

aprovada em outubro de 2006.



---

Prof. Luci Pirmez, D.Sc., COPPE/UFRJ, Brasil

Em memória do meu querido e amado pai.

## **AGRADECIMENTOS**

À minha mãe, mulher maravilhosa que sempre incentivou todos os seus filhos.

À minha mulher e filhas pela compreensão e carinho.

Ao meu irmão e irmã, apenas por tudo que passamos juntos.

Aos companheiros do MOT-CN turma 2004 em especial a Jaime, Roberto, Bárbara, Pedro, Paulo, Oscar e Rui por tudo que passamos juntos e pelo laço de amizade criado.

A todos os professores(as) do MOT-CN pela dedicação aos seus alunos.

A todos os funcionários(as) do MOT-CN pelo apoio prestado.

## **RESUMO**

Lima, Carlos Arthur Rodrigues. GERENCIAMENTO EM GRID. Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2005.

No início da década de 80 surgia a computação pessoal reduzindo custos e possibilitando a disseminação dos computadores em empresas de pequeno e médio porte, inclusive nas residências. Sistemas de grande porte, os ditos mainframes, eram exclusividade de grandes empresas, instituições de pesquisa e militares. No âmbito acadêmico, pesquisas avançadas necessitavam de maior poder computacional, não somente no que tange ao processamento, mas também à armazenagem de informações, mas isto ainda representava um custo muitas vezes proibitivo.

Em meados da década de 90 surgiu o termo “the Grid”, que define uma infraestrutura de computação distribuída utilizada, principalmente, pela comunidade acadêmica para pesquisas avançadas. Esta proposta tem como finalidade a utilização de recursos ociosos, tais como processamento e armazenamento entre outros, de centenas ou milhares de computadores. Este dado resultou no desenvolvimento de ferramentas de gerenciamento, tanto dos processos quanto das redes envolvidas. Diversas propostas surgiram neste contexto e este trabalho é uma tentativa de apresentar algumas das mais populares destas ferramentas e traçar um parâmetro entre elas.

## **ABSTRACT**

Lima, Carlos Arthur Rodrigues. GERENCIAMENTO EM GRID. Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2005.

Begins of the 80 years the personal computation it appeared reducing costs and making possible the introduction of the computers in companies of small and medium load as well as in our houses. Systems of great load, the said mainframes, were exclusiveness of great companies, research institutions and military institution. In the academic ambit, advanced researches needed larger power computacional, not just in what it plays to the processing as well as the storage of information, more this still represented a cost a lot of times prohibitive.

In the middle of the decade of 90 the term " the Grid " appeared that denoted an infrastructure of distributed computation used mainly by the academic community for advanced researches. This proposal not just uses the idle resource of hundreds or thousands of computers as well as the capacity of storage of the same ones. This fact not just took to the development of administration tools of processes as well as of the involved nets. Several proposed they will appear in this context and this work is an attempt of showing some of the most popular of these tools as well as to trace a comparative one among the same ones.

## LISTA DE FIGURAS

	Página
Figura 1 – Organização Virtual	16
Figura 2 – Estrutura I-WAY	30
Figura 3 – Globus Toolkit	31
Figura 4 – Mecanismo de comunicação Nexus	33
Figura 5 – Estrutura MDS	34
Figura 6 – Mecanismo RIO	35
Figura 7 - Condor-G	37



## LISTA DE ABREVIATURAS E SIGLAS

VO	Organização Virtual
PAD	Processamento de Alto Desempenho
QoS	Qualidade de Serviço
SSL	Secure Sockets Layer
GSS	Generic Security System
API	Application Programming Interface
MDS	Metacomputing Directory Service
LDAP	Lightweight Directory Access Protocol
NFS	Network File System
DFS	Distributed File System
ADIO	Abstract-Device Interface for I/O
RIO	Remote I/O for Metacomputing Systems
GGF	Global Grid Fórum
OGSA	Open Grid Service Architecture
OGSI	Open Grid Service Infrastructure
XML	Extensible Markup Language
WSDL	Web Services Description Language
WS-RF	Web Service Resource Framework
CORBA	Common Object Request Broker Architecture
RMI	Remote Method Invocation

# SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>11</b>
1.1 OBJETIVOS .....	12
1.2 ORGANIZAÇÃO DO TRABALHO.....	12
<b>2 GRID COMPUTACIONAL .....</b>	<b>13</b>
2.1 CONCEITO.....	13
2.2 ORGANIZAÇÃO VIRTUAL – VO .....	14
2.3 DISTRIBUIÇÃO DE PROCESSOS NUM GRID.....	17
2.4 CLUSTER X GRID .....	20
2.5 BENEFÍCIOS DO GRID .....	21
<b>3 CONVERGÊNCIA ENTRE WEB SERVICES E GRIDS – OGSF E WS-RF .....</b>	<b>22</b>
<b>4 FERRAMENTAS DE GERENCIAMENTO .....</b>	<b>29</b>
4.1 GLOBUS TOOLKIT .....	29
4.1.1 Breve historico.....	29
4.1.2 Descrição dos módulos Globus.....	32
4.2 CONDOR .....	35
4.2.1 Definição .....	35
4.2.2 Condor g .....	36
4.2.3 Ferramentas Condor .....	37
<b>5 METODOLOGIA UTILIZADA .....</b>	<b>39</b>
<b>6 CONCLUSÃO FINAL E POSSÍVEIS TRABALHOS FUTUROS .....</b>	<b>40</b>
<b>7 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>41</b>

## 1 INTRODUÇÃO

O gerenciamento de toda a infra-estrutura que compõe o chamado Grid computacional, quais sejam os hardwares e softwares utilizados, é na atualidade um grande desafio na área de computação distribuída.

O termo “Grid” ganhou força em meados da década de 90 com inspiração nas malhas de interligação dos sistemas de energia elétrica. Este termo dá significação a uma infra-estrutura de computação distribuída que utiliza computadores ociosos ou com algum ciclo ocioso de cpu, seus meios de armazenamento, bem como as diversas redes heterogêneas utilizadas para interligá-los.

Em 1961, numa palestra em comemoração ao centésimo aniversário do MIT (Massachusetts Institute of Technology), John MacCarthy [1] foi quem primeiro sugeriu uma tecnologia de compartilhamento de recursos, que segundo ele poderia ser organizada como um serviço público, tal como as redes de telefonia.

Atualmente, os computadores de uso pessoal possuem grande capacidade de processamento e armazenamento, muitos deles com alto grau de ociosidade, e esta condição os caracteriza como máquinas propensas a compor um Grid.

É necessário que se tenha em mente que o Grid é um caso particular da computação distribuída. A partir da existência de uma infra-estrutura física e lógica, onde exista a possibilidade de coordenar os trabalhos que serão processados e garantir a qualidade do serviço, esta passa a ser um Grid.

## **1.1 OBJETIVOS**

O objetivo deste trabalho é apresentar algumas das atuais ferramentas de gerenciamento de um Grid.

## **1.2 ORGANIZAÇÃO DO TRABALHO**

A presente monografia contém sete capítulos. No segundo capítulo são apresentados os conceitos básicos de Grid Computacional, Organizações Virtuais, Distribuição de Processos, bem como é efetuada uma comparação entre Grid e Cluster. O terceiro capítulo aborda a convergência entre os serviços Web e os Grid Services. No quarto capítulo são apresentadas duas das mais populares ferramentas para o gerenciamento do Grid. O quinto capítulo apresenta a metodologia utilizada neste trabalho e o sexto capítulo à conclusão e possíveis trabalhos futuros. O sétimo capítulo é reservado a referencia bibliográfica.

## 2 GRID COMPUTACIONAL

### 2.1 CONCEITO

O conceito de Grid Computacional nos remete ao modelo de distribuição de energia elétrica utilizado nos Estados Unidos por volta do ano de 1910, chamado de *power grid*. Nele, um dispositivo utiliza a energia elétrica distribuída por uma grande malha sem dar importância à origem desta geração de energia.

Um Grid Computacional é composto por uma infra-estrutura de hardware e software que prove dependência, consistência, penetração e um baixo custo de acesso a uma grande capacidade computacional, sem se importar onde esta capacidade computacional é disponibilizada. Fica claro então, que o Grid Computacional é na realidade uma estrutura que tem como objetivo o compartilhamento de recursos, cujo acesso aos mesmos possam ser feitos de qualquer local e a qualquer hora.

Para tentar deixar mais clara a definição do que é um Grid, Ian Foster do Laboratório Nacional Argonne [2], sugeriu uma lista onde as diversas definições do que é um Grid estão sintetizadas. Esta lista possui três pontos, que caso se apliquem a um suposto Grid, o caracterizam como tal. De acordo com o que consta nesta lista, um Grid possui recursos coordenados que não se submetem a um controle centralizado; utiliza padrões abertos e com interfaces e protocolos de propósito geral e provem o mínimo em qualidade de serviços.

Inicialmente usado de forma quase exclusiva pela comunidade acadêmica, que necessitava de processamento de alto desempenho (PAD), sendo este o seu berço, hoje, empresas como IBM, SUN e outras, vêem o Grid como uma forma de

oferecer uma nova proposta de serviços e produtos aos seus clientes, reduzindo os custos e aumentando a produtividade.

## **2.2 ORGANIZAÇÃO VIRTUAL – VO**

Existe certa complexidade para a conceituação do que é uma Organização Virtual (VO) e vários autores utilizam conceitos próprios para tentar explicá-la. Byrne [3] define uma Organização Virtual como sendo “uma rede temporária de empresas independentes, unidas por tecnologias de informação que partilham competências técnicas, custos e acesso aos mercados de cada um. As empresas unem-se para explorar oportunidades de mercado específicas e posteriormente, separam-se”.

Os autores Jansen, Steenbakkers e Jägers [4] afirmam que “uma Organização Virtual é uma combinação de várias partes (pessoas e/ou organizações) dispersas geograficamente, que partilham as suas competências e recursos, com o intuito de alcançar um objetivo comum. Os parceiros desta organização têm igual estatuto e estão dependentes de ligações eletrônicas para a coordenação das suas atividades”. Segundo Mowshowitz [5], “a essência das Organizações Virtuais é a metagestão da atividade de atingir objetivos de forma independente da sua realização. Metagestão é a gestão de uma atividade organizada virtualmente. Esta atividade contém um conjunto de requisitos, um conjunto de empresas que os podem satisfazer e uma tabela de procedimentos que mapeia as empresas nos requisitos”.

Existem várias linhas de pensamento para definir uma Organização Virtual e na tentativa de resolver este problema e chegar a um denominador comum quanto a esse conceito, Bultje e Van Wijt [6] pesquisaram na literatura existente e

encontraram vinte e sete características que definem uma VO, sendo que seis delas são consideradas imprescindíveis para chegarmos a uma correta definição. Uma Organização Virtual pode ser baseada nas **competências nucleares**, onde os parceiros contribuem apenas com o que têm de mais forte. A união de todas as competências nucleares permite atingir o objetivo maior, que é o de ter uma organização excelente, onde cada membro formador contribua com a sua melhor porção. Essa Organização pode ser formada por uma **rede de organizações independentes**, onde as várias empresas formadoras são ligadas por relações semi-estáveis. Deve possuir uma **identidade própria** e cada membro formador deve ter sua identificação visível. Deve ainda ser **baseada nas tecnologias de informação**, sendo que, a postura de quase todos os autores, converge para o fato de que a mesma deve ser construída a partir de computadores. **Não deve possuir hierarquia**, sendo que cada membro formador deve estar igualitário na organização, crescendo a eficiência e decrescendo os gastos. Uma VO precisa primar por uma **separação entre os níveis estratégico e operacional**, havendo uma clara distinção entre os requisitos abstratos e a implementação concreta para atingir os objetivos organizacionais.

A definição concludente é a de que uma Organização Virtual é composta por várias instituições independentes, dispersas geograficamente que utilizam uma estrutura de tecnologia, permitindo a sua união com uma missão parcialmente sobreposta. Os membros desta organização participam com as suas competências primordiais e a cooperação é fundada em relações semi-estáveis. Os produtos e serviços fornecidos estão fortemente voltados para o usuário final e muito dependente das oportunidades de negócio e de inovação. Pode ainda ser caracterizada por ter uma identidade própria, por exigir lealdade entre os parceiros,

cooperação baseada na confiança e nas tecnologias de informação e por manter uma distinção nítida entre os níveis operacional e estratégico.

Em The Anatomy of The Grid [7], o conceito de Organização Virtual é inserido no mundo da computação distribuída. Fica mais simples entender este conceito quando pensamos em participantes que desejem compartilhar recursos para a solução de uma tarefa, sendo que este compartilhamento pode envolver desde a troca de um simples documento até o acesso remoto a softwares, ciclos de cpu, dados e repositórios para armazenamento de informações, entre outros.

Uma VO é responsável pela forma e pelo momento em que os seus recursos são disponibilizados. Um mesmo recurso pode ser utilizado de formas diferentes, isto é, pode-se definir numa Organização Virtual que um computador a ela pertencente será utilizado para executar uma tarefa específica quando a requisição partir de um determinado arranjo de compartilhamento, porém, a mesma máquina, pode executar uma tarefa de computação genérica quando a requisição partir de um outro arranjo. A figura abaixo dá uma melhor idéia do exposto.

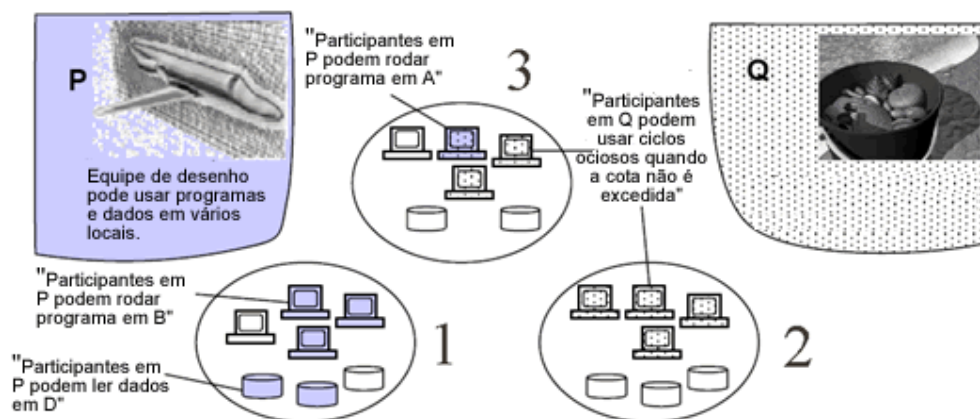


Figura 1 – Organização Virtual



Na figura 1, temos três organizações, 1, 2 e 3, e duas VO's, P e Q. Pode-se ver que a organização 1 utiliza recursos da VO P e a organização 2 utiliza recursos da VO Q. A organização 3 utiliza recursos das duas VO's, P e Q

## **2.3 DISTRIBUIÇÃO DE PROCESSOS NUM GRID**

Em um Grid, os recursos disponibilizados podem ser os mais variados possíveis, sendo que quando se pensa apenas no processamento de dados, o compartilhamento de ciclos de cpu é o mais usual. Pode-se tomar como exemplo o projeto SETI@Home [8]. Neste projeto, um pequeno software de análise espectral é instalado em um computador local e fragmentos de sinal do rádio telescópio de Arecibo são recebidos via Internet. O software em questão, dependendo de como a configuração do mesmo é definida, pode utilizar ciclos ociosos de cpu, funcionando como uma proteção de tela no computador hospedeiro, ou ser executado o tempo todo em segundo plano. Após o término da análise, o resultado é enviado a um nó central, via Internet, que se encarrega de reunir todos os resultados enviados pelos milhares de computadores que participam do projeto, de forma voluntária. Apenas como exemplo do poder de processamento do Grid formado no projeto Seti@Home, tem-se que o mesmo computa dados em média a uma velocidade de 10 TeraFlops.

Podemos enumerar duas formas de aproveitamento dos ciclos ociosos de cpu. Na primeira, pode ser usada a divisão do processamento para que o ciclo seja executado em vários nós de forma paralela, e na segunda, o processamento é executado de forma exclusiva em um único nó. No primeiro caso, a aplicação exige características que permitam a sua divisão, para que a mesma possa ser processada de forma paralela nos diversos nós por ela utilizada. Bibliotecas

específicas têm sua utilização em conjunto com o sistema de gerência do Grid para prover total abstração da aplicação, no que tange à distribuição dos processos pelos nós utilizados. No segundo caso, a utilização de um único nó do Grid para o processamento de uma aplicação, se justifica quando o nó remoto possuir hardware superior ao hardware local, aumentando com isso a velocidade de processamento.

Um Grid é um ambiente heterogêneo em termos de hardware e sistema operacional e por isso é necessário que haja um grande cuidado na escolha dos nós onde a aplicação será processada. Em um Grid de grandes proporções, torna-se difícil que um usuário tenha conhecimento de todos os recursos disponíveis sobre os nós formadores deste Grid. Para tal, algumas ferramentas de gerência possuem escalonadores que realizam a tarefa de seleção de nós adequados ao processamento de uma determinada aplicação, tornando a infra-estrutura do Grid transparente para o usuário.

A existência de uma rede interligando diversos computadores, não significa que seus recursos possam ser aproveitados para a criação de um Grid onde tarefas serão executadas. Alguns fatores, como o nível de utilização dos equipamentos, a disponibilidade de ciclos ociosos de cpu e a estrutura da rede utilizada na interligação, podem inviabilizar a formação do Grid. As diversas máquinas possíveis formadoras do Grid devem ser analisadas para que seja possível determinar quanto de ociosidade de cpu estará disponível em cada uma delas. É importante salientar que o perfil de utilização de uma determinada máquina pode mudar ao longo do tempo, tornando-a inviável para a utilização no Grid. Faz-se necessário refletir também, que uma máquina que participa de um Grid de forma não exclusiva, oferecendo, por exemplo, recurso de processamento por ociosidade de ciclo de cpu, pode não ser uma boa candidata ao Grid se possuir picos de utilização, pois, a cada

pico, o processamento externo deve ser interrompido ou migrado para outro nó, dando prioridade à execução da tarefa local.

Outro fator que também deve ser levado em consideração é o fato de que uma máquina que apresenta ciclos ociosos de cpu e não possui poder de processamento para executar as tarefas que lhe possam ser atribuídas em tempo hábil, pode não ser uma escolha adequada a participar do Grid. Testes de desempenho devem ser executados nos diversos computadores candidatos a formar o Grid e os resultados obtidos podem ser disponibilizados para o escalonador, de forma que ele possa promover a escolha dos nós que irão executar uma determinada tarefa, baseado nas características de processamento de cada nó. A obtenção destas informações pode ser feita de forma automática ou de forma prévia. As duas formas apresentam vantagens e desvantagens.

Na forma automática, um novo nó pode ser inserido ao Grid e as características deste nó serão coletadas e divulgadas, por meio de rotinas específicas, para os escalonadores do Grid, porém, a falta de exatidão das características do novo nó representa uma desvantagem.

Na forma prévia, a inserção de um novo nó deve ser feita através da coleta de informações sobre as suas características e os resultados obtidos devem ser colocados em arquivos de configuração para cada máquina. Isto garante uma maior exatidão nas medidas coletadas, porém, caracteriza uma desvantagem, pois as características de cada máquina devem ser obtidas manualmente.

Outro ponto importante é o fato de que não há ganho de desempenho em um Grid se os processos distribuídos nos vários nós fizerem uso intensivo da estrutura de rede usada para interligação, pois, o tempo de comunicação da rede pode se tornar um afunilamento considerável. Deve-se observar também que não haverá

sentido em distribuir processos entre os nós de um Grid, se o tempo para esta operação for superior ao tempo usado no processamento remoto.

## **2.4 CLUSTER X GRID**

Quando se pensa que tanto um Grid quanto um Cluster são formados por diversos computadores interligados entre si, por intermédio de uma estrutura de comunicação de dados, é comum que os dois termos se confundam. Apesar do fato de as duas arquiteturas possuírem a interligação entre os diversos computadores que as compõem via rede de dados, as duas estruturas se diferenciam uma da outra. Um Cluster se caracteriza por ser uma estrutura local formada por hardware e software conectados entre si através de uma rede de dados gerenciados por estrutura única e central, onde os diversos nós formadores do Cluster operam de forma cooperativa para a solução de uma determinada tarefa. Um Grid, em oposição ao conceito de Cluster, opera de forma geograficamente distribuída, não submetido a um gerente central e o gerenciamento dos diversos nós que o compõe, fica a cargo de cada VO participante e para que um usuário tenha acesso aos recursos disponibilizados pelo Grid, estes devem ser requisitados.

Vemos desta forma que os Grid's, se comparados aos Clusters's, são sistemas heterogêneos e distribuídos. Pelo fato de estarem seus diversos VO's formadores dispersos por áreas com grande abrangência geográfica, as aplicações que de uma forma utilizam menos os recursos de rede são as mais indicadas para a utilização do Grid. As redes utilizadas para a interligação das diversas VO's formadoras do Grid poderão apresentar características, que de certa forma, irão causar impacto negativo no fluxo de informações.

## **2.5 BENEFÍCIOS DO GRID**

Um Grid Computacional é na realidade uma estrutura que permite o compartilhamento de recursos, dedicados ou ociosos e fica fácil entender que sua implementação é indicada para diversas áreas. Através da implantação de uma estrutura de Grid, uma organização que possui alcance global pode agregar recursos, tornando-os disponíveis para a organização como um todo. Bases de dados com TeraBytes de informação, podem ser armazenadas, acessadas e compartilhadas de forma remota pelos diversos nós formadores do Grid. Mesmo com a utilização de computadores de pequeno porte, o poder computacional de um Grid atinge TeraFlops, pois neste caso a união dos diversos nós formadores do Grid respondem pela velocidade de processamento global do Grid.

### 3 CONVERGÊNCIA ENTRE WEB SERVICES E GRIDS – OGSÍ E WS-RF

Um dos motivos de crítica por parte dos diversos usuários de um Grid, refere-se ao fato de que as ferramentas disponibilizadas por ele, muitas vezes são de grande complexidade, tanto na instalação e configuração como na sua utilização.

É comum que um usuário necessite utilizar uma pequena parcela dos diversos serviços disponibilizados num *middleware* de Grid, porém, a utilização desses serviços torna-se extremamente pesada devido ao simples fato de não ser possível a utilização de uma determinada ferramenta requerida para a realização de uma tarefa, já que o Grid disponibiliza um conjunto completo de recursos.

O Grid Computacional, antes utilizado exclusivamente no meio acadêmico, está se tornando uma ferramenta popular, ou seja, de utilização geral. Entende-se então, que a “limitação” descrita no parágrafo anterior também deve ser estendida a qualquer outro tipo de utilizador, o que levou aos grupos que pesquisam e desenvolvem ferramentas e recursos de Grid, a buscar uma padronização que pudesse satisfazer aos dois mundos.

Um dos vários grupos de pesquisa de Grids, o GGF – Global Grid Forum [9] foi o responsável pela especificação de uma arquitetura de serviços, a OGSA – Open Grid Service Architecture, que possui como ponto central os chamados Grid Services. É OGSA que define a infra-estrutura de serviços utilizada por um Grid chamada de OGSÍ – Open Grid Service Infrastructure, o qual descreve as funções presentes em um Grid Service e que não são encontradas em Web Services tradicionais. OGSA também é responsável pela padronização dos diversos serviços que são encontrados em um Grid por meio da especificação de um conjunto de interfaces própria destes serviços.

Um dos pontos principais de um Grid é o anonimato, ou seja, um usuário não precisa saber quem ou o que está prestando a ele um determinado tipo de serviço ou provendo um recurso. Para utilizar este grau de abstração, qualquer arquitetura de sistema distribuída como Corba, RMI ou outros, poderia ser utilizada, porém, por ser um sistema fracamente acoplado, a opção lógica foi a de se utilizar a arquitetura de serviços Web. Num Web Service é clara a separação entre a interface, que é escrita em XML e chamada de WSDL – Web Services Description Language e a implementação.

A arquitetura Web Service foi criada para prover algum tipo de serviço para a Web. Ela não satisfazia integralmente os padrões definidos pela OGSA e, portanto, sofreu adequações e extensões que originaram os Grid Services. Grid Services são na realidade, serviços Web que seguem os padrões definidos na OGSA. OGSA flexibiliza a utilização de suas interfaces por definir um baixo acoplamento entre o cliente e o provedor de um serviço. Assim sendo, é possível mudar a implementação de um determinado serviço sem que para tanto seja necessária uma mudança da implementação num cliente. O inverso também é verdadeiro.

OGSA define o modelo de arquitetura e identifica serviços básicos, porém, não define a infra-estrutura destes serviços. Para tanto, uma nova especificação a OGSF – Open Grid Service Infrastructure, foi liberada em Julho de 2003 pelo GGF [9] e tem por finalidade definir um modelo de componente usando as extensões WSDL e esquemas XML, para apresentar os conceitos de Web Service stateful. Define também aproximações para a criação, nomeação, gerenciamento do tempo de vida de instâncias de serviços, notificação assíncrona de mudanças no estado de serviço, gerenciamento e representação de instâncias de serviços, manipulação de erros e a forma como um cliente deve interagir com o Grid Service.

De um modo geral a especificação OGSi define:

- Um conjunto de extensões para a WSDL baseado na WSDL 2.0;
- Padrões de estrutura e operação para representação, consulta e atualização de dados sobre os serviços;
- Estruturas de manipulação e referência, usadas para endereçar os Grid Services;
- Informação sobre falhas sem a modificação do modelo original WDSL;
- Conjunto de operações que possibilitam a criação e a destruição de Grid Services;
- Mecanismos de notificação assíncrona para o caso de haver mudança de valores dos dados.

Mesmo com todas as facilidades apresentadas, o modelo definido pelo OGSi recebeu críticas da comunidade Web Service. Na realidade o OGSi apresenta algumas desvantagens que impediriam a convergência dos padrões. São elas:

- A especificação OGSi é extremamente longa e a mesma não possui uma clara separação das funções que dão suporte à adoção incremental;
- OGSi não funciona bem com as ferramentas Web Services e XML existentes;
- Na versão 1.0, OGSi utiliza de forma agressiva esquemas XML que não utilizam, necessariamente, construções padrão;
- A versão 1.0 é muito voltada para a orientação a objeto, modelando um recurso stateful como um serviço Web. Por este motivo, alguns puristas argumentarão que Web services não têm estados ou instâncias e além disso, Web Services não implementam serviços dinâmicos de criação e destruição de recursos;



- Por ter sido construída a partir de rascunhos da especificação WSDL 2.0 e devido a atrasos na divulgação desta versão, as ferramentas e serviços que utilizam a WSDL 1.0 se tornaram incompatíveis com a OGSi.

Em virtude destes problemas e pela evolução natural da arquitetura Web Service, um novo padrão foi apresentado com o intuito de promover a convergência entre os Web Services e Grids Computacionais. Este novo padrão foi chamado de Web Service Resource Frameworks (WS-Resource Frameworks ou WS-RF) e diferentemente do modelo proposto pelo OGSi, foi baseado diretamente em Web Services e não em Grid Services. Na realidade, WS-RF é basicamente, o resultado do refinamento da OGSi com o objetivo de aproveitar a existência dos novos padrões que surgiram para Web Services, tais como WS-Addressing, WS-Notification e assimilar a demanda da comunidade Web Services.

As principais preocupações do Ws-Resource Frameworks são com a criação, endereçamento, tempo de vida e inspeção dos recursos stateful. WS-RF possui cinco especificações técnicas e mais uma especificação de notificação que são:

a) WS-ResourceLifetime – O tempo de vida de um recurso é definido como o período entre sua criação e a sua destruição. WS-ResourceLifetime padroniza os meios pelos quais um recurso pode ser destruído e também define a forma pela qual o tempo de vida de um recurso pode ser monitorado. São definidas duas formas de destruição de um recurso: imediatamente ou por agendamento. A destruição imediata de um recurso pode ser realizada através da troca de mensagens específica para tal. Define também a maneira pela qual um recurso pode ser destruído após um período de tempo, seja por inatividade, ou pelo fato de o cliente

ter-se desconectado do Grid. Neste caso o recurso se auto-destruirá depois de determinado período, sem que o cliente o tenha requisitado.

b) WS-ResourceProperties – Inspirada em uma parte da OGSF, padroniza a forma pela qual as propriedades de um WS-Resource podem ser declaradas como parte de uma interface de serviços Web. Também padroniza um conjunto de mensagens trocadas, utilizando WSDL e XML que permitem que um cliente pesquise e atualize os valores de um recurso.

c) WS-RenewableReferences – Define um mecanismo que deve ser utilizado para recuperar um marcador de “endpoint” que tenha se tornado inválido. Este mecanismo pode ser aplicado a qualquer marcador de “endpoint”, mas é particularmente útil no caso deste “endpoint” se referir a um WS-Resource, como também pode prover uma referência estável e persistente para um WS-Resource, que é acessado várias vezes e que deve preservar o mesmo estado com o passar do tempo.

d) WS-ServiceGroup – Define os meios pelos quais os serviços Web e os WS-Resources podem ser agrupados e agregados para um propósito específico. Esta especificação pode ser utilizada para organizar coleções de WS-Resources tais como construtores de registro ou construtores de serviços, que podem executar operações em uma coleção de WS-Resources. Grupos podem ser definidos como uma coleção de membros com mesmo propósito. Também define a interface utilizada para o gerenciamento dos membros de um ServiceGroup.

e) Ws-BaseFaults – Um projetista de aplicações Web service normalmente utiliza interfaces criadas por outros. Porém, o gerenciamento das mensagens de erro torna-se difícil quando cada interface utiliza uma forma de representar um tipo de erro que muitas vezes é comum, diferindo apenas na forma pela qual é apresentado.

WS-BaseFaults define um tipo de esquema XML que padroniza a maneira pela qual estes erros são utilizados por Web services.

f) WS-Notification – Forma uma “família” separada das especificações da WS-RF e descreve as regras básicas, conceitos e padrões necessários para permitir que um cliente registre interesse em receber mensagens de notificação dando ciência sobre mudanças de estado de um recurso. Uma mensagem de notificação pode indicar uma mudança na propriedade de um recurso ou qualquer mudança interna no estado ou ambiente de um servidor de recurso. Um cliente registra o seu interesse em receber uma ou mais notificações por meio de uma mensagem de requisição “subscriber” e em resposta, recebe uma mensagem do tipo WS-Resource-Qualified referente à subscrição. O relacionamento entre um cliente subscritor e o produtor de uma mensagem é gerido pelo Ws-Resource que usa o WS-ResourceProperties e o WS-ResourceLifeTime como auxiliares neste gerenciamento. Uma segunda especificação de notificação, dentro da WS-Notification, é a WS-Topics, que organiza as mensagens de notificação de forma hierárquica, permitindo a um cliente entender de forma conveniente quais mensagens de notificação estão disponíveis para a subscrição. Existe ainda uma terceira especificação dentro da WS-Notification que é a WS-BrokeredNotification que é utilizada para implementar serviços de gerência para outras entidades que produzem mensagens de notificação.

Os padrões WS-RF e OGSF possuem propostas iguais, ou seja, promover a convergência entre Grids Computacionais e a tecnologia de serviços Web, diferindo apenas na forma como são implementados. OGSF utiliza Grids Services enquanto WS-RF utiliza Web Services para a mesma finalidade. A grande diferença entre os

dois padrões diz respeito ao fato de que a WS-RF utiliza melhor os padrões XML existentes e também, diferentemente do que aponta OGSF, possui uma construção mais simples do que ele. Não se pode afirmar, no entanto, que WS-RF esteja de todo completo, isto é, muito ainda há que se fazer para que as restrições não resolvidas ou ainda agravadas sejam solucionadas a contento.

## **4 FERRAMENTAS DE GERENCIAMENTO**

Dentre as diversas ferramentas que surgiram nos últimos anos para suporte e gerenciamento de um Grid Computacional, destacaremos duas: Globus, que é a ferramenta que talvez cause o maior impacto, sendo quase um padrão para a criação e o gerenciamento de um Grid, servindo inclusive de base para outras ferramentas que utilizam alguns de seus recursos e Condor, que permite uma integração entre múltiplos domínios.

### **4.1 GLOBUS TOOLKIT**

#### **4.1.1 BREVE HISTORICO**

Em 1994, Rick Stevens do Argonne National Laboratory e Tom DeFanti do Electronic Visualization Laboratory da Universidade de Illinois, em Chigago, propuseram a conexão de supercomputadores e outros recursos espalhados por 17 (dezessete) diferentes localidades na América do Norte, utilizando redes de alta velocidade para a criação de um Grid de abrangência nacional. Este experimento, batizado de I-WAY [10], foi iniciado duas semanas da antes da conferência de super-computação de 1995 e mantido durante a mesma. Ian Foster [10] do Argonne National Laboratory, liderou um pequeno grupo que criou os protocolos necessários para que os usuários pudessem executar aplicações em computadores dispersos pelo país.

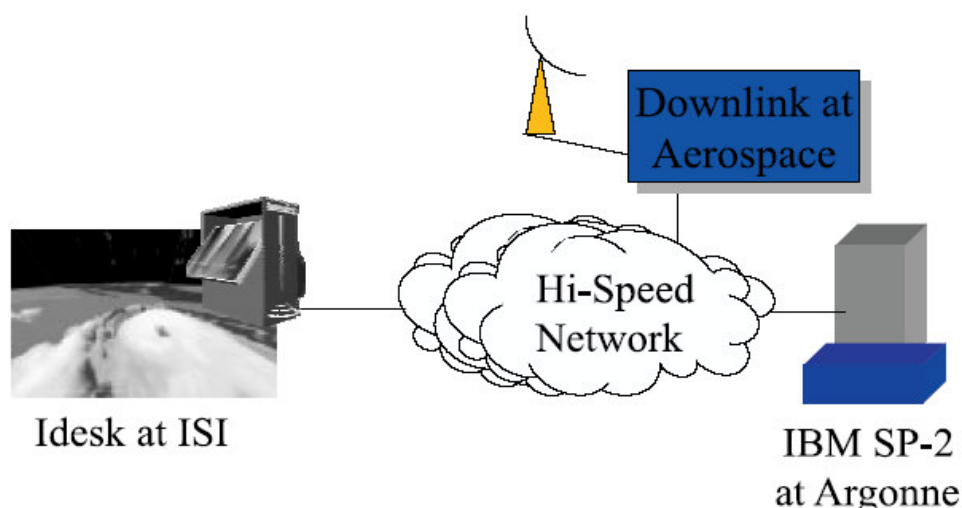


Figura 2 – Estrutura usada pelo I-WAY – O Downlink do satélite estava localizado em El Segundo, Califórnia, o supercomputador no Argonne e o dispositivo gráfico de visualização em Los Angeles.

O sucesso deste experimento deu origem ao projeto Globus [11] que hoje se apresenta na quarta versão (quatro).

Um importante conceito usado no projeto Globus é o da Metacomputação. Um metacomputador é na realidade um supercomputador virtual construído de forma dinâmica e geograficamente distribuída, sendo que a ligação entre os diversos nós, que são usados para a formação do mesmo, é feita através de redes de alta velocidade. A Metacomputação é motivada pela necessidade de acessar recursos não localizados dentro de um mesmo computador.

Da mesma forma que num sistema distribuído, um Grid Computacional deve conectar-se através de uma rede de dados com uma extensa gama de recursos que normalmente se localizam em diferentes domínios administrativos. Por outro lado, caso o objetivo seja performance, as interfaces tanto quanto os aplicativos são radicalmente diferentes se comparados aos utilizados nos modelos distribuídos.

Como num sistema paralelo, aplicações para meta-computação devem cuidadosamente fazer a requisição para que os recursos necessários sejam encontrados e alocados. Vê-se que a meta-computação pode usufruir da mesma tecnologia aplicada na construção de softwares usados por sistemas paralelos e distribuídos, contudo requer avanços em técnica e nas ferramentas de desenvolvimento de software utilizadas para este fim.

Uma infra-estrutura de meta-computação deve fornecer as interfaces e as capacidades básicas em áreas como comunicação, informação, localização, agendamento de recursos, autenticação e acesso a dados. A junção de todos estes componentes forma a chamada máquina abstrata de meta-computação que pode ser aplicada a uma grande gama de infra-estruturas, serviços e aplicações.

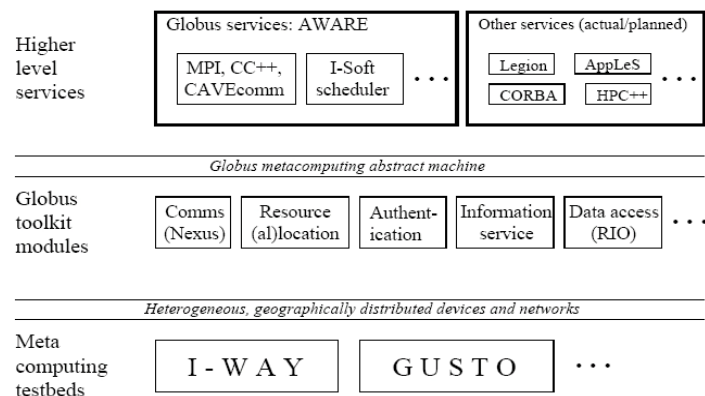


Figura 3 – Globus Toolkit

Globus é um conjunto de módulos onde cada modulo define uma interface de alto nível utilizada para invocar os mecanismos necessários para o módulo em questão e provê implementações, as quais utilizam recursos de baixo nível, necessários para a implementação destes mecanismos nos mais variados ambientes.

#### 4.1.2 DESCRIÇÃO DOS MÓDULOS GLOBUS

Globus Toolkit fornece seis módulos: alocação e localização de recursos, comunicação, autenticação, criação de processos, informação unificada de recursos e acesso a dados.

- Alocação e localização de recursos – provê mecanismos para que uma aplicação requisiite recursos, identifique um tipo de recurso quando uma requisição é feita e agende a utilização de um recurso, assim como sua localização. O mecanismo de localização de recurso é necessário, pois, uma aplicação não sabe onde um determinado recurso está disponível. A alocação de recursos envolve desde o agendamento para a utilização do mesmo até a iniciação de processos necessários para a realização de uma tarefa como, por exemplo, acesso a dados.

- Comunicação – este componente fornece mecanismos básicos para a comunicação. Permite uma implementação eficiente de uma grande variedade de métodos de comunicação, incluindo passagem de mensagens, chamada remota a procedimentos, distribuição de memória compartilhada e multicast. Este mecanismo deve conter conhecimentos sobre os parâmetros de QoS da rede tais como Jitter, latência e largura de banda.

O módulo de comunicação do Globus é baseado na biblioteca de comunicação Nexus que define as seguintes abstrações: contexto, link de comunicação de processos e requisição remota de serviços. As funções do Nexus que manipulam estas abstrações formam a interface de comunicação Globus, sendo que esta é intensamente utilizada por outros módulos do Globus e também para a construção de outros serviços de alto nível como, por exemplo, ferramentas de programação paralela.



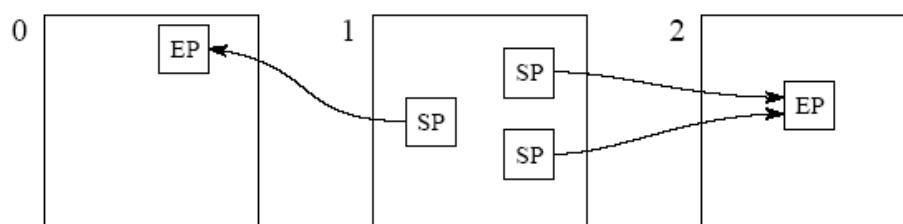


Figura 4 – Mecanismo de comunicação Nexus – Três contextos e três links de comunicação. Três startpoints (SP) em 1 ligam-se a endpoints em 0 e 2.

- Autenticação – este mecanismo fornece componentes básicos que são aplicados na validação da identidade, tanto do usuário quanto dos recursos, sendo também o responsável pela implementação de outros serviços relacionados à segurança, como a autenticação e a segurança dos dados. Nas versões iniciais do Globus eram aceitas senhas Unix RSH, e também autenticação SSL, entretanto, para que fosse atingido um alto grau de abstração passou-se a utilizar o GSS, Generic Security System que prove padrões e API's para a obtenção de credenciais, senhas e certificados, para a autenticação, tanto do cliente quanto dos servidores. Pode ser utilizado em conjunto com diferentes métodos de segurança tais como Kerberos e SSL.

- Informação unificada de recursos – é o mecanismo responsável por manter informações sobre a estrutura formadora do Grid, assim como do seu status. Deve permitir que componentes postem e recebam informações. Chamado de MDS, Serviço de Diretório de Metacomputação, é construído dentro dos padrões usados no protocolo LDAP, guardando informações de interesse geral sobre os recursos disponíveis no Grid.

```

GlobusHost OBJECT CLASS
SUBCLASS OF GlobusResource
MUST CONTAIN {
    hostName      :: cis,
    type          :: cis,
    vendor        :: cis,
    model         :: cis,
    OStype        :: cis,
    OSversion     :: cis
}
MAY CONTAIN {
    networkNode   :: dn,
    totalMemory   :: cis,
    totalSwap     :: cis,
    dataCache     :: cis,
    instructionCache :: cis
}

GlobusResource OBJECT CLASS
SUBCLASS OF top
MUST CONTAIN {
    administrator :: dn
}
MAY CONTAIN {
    manager       :: dn,
    provider      :: dn,
    technician    :: dn,
    description   :: cis,
    documentation :: cis
}

```

Figura 5 – exemplo de MDS GlobusHost e GlobusResource

- Criação de processos – é usado para iniciar o processamento desde que o recurso ou os recursos necessários já tenham sido localizados e devidamente alocados. Inclui a criação do ambiente de execução, a passagem de argumentos, a integração de novos processos, iniciação, execução e finalização de um processo.

- Acesso a dados – responsável por fornecer acesso remoto em alta velocidade a dispositivos de armazenamento localizados em múltiplos domínios administrativos. Sistemas de arquivo distribuídos como o NFS e o DFS podem realizar tal tarefa, no entanto, não foram desenhados para aplicações que necessitem de alta-performance. Para resolver o problema, o módulo de acesso a dados do Globus define as premissas que serão utilizadas no acesso remoto a sistemas de arquivo paralelo. Baseado no ADIO que define as interfaces de abertura, fechamento, leitura e gravação de dados em sistemas de arquivo paralelo, o RIO é utilizado no Globus como interface de acesso a dados remoto. RIO utiliza recursos do NEXUS

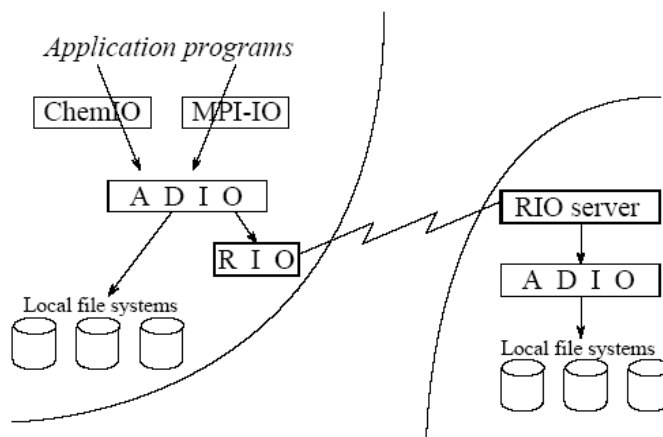


Figura 6 – RIO mecanismo para acesso remoto com alta-performance

## 4.2 CONDOR

### 4.2.1 DEFINIÇÃO

Condor [14] é um sistema que busca alta vazão e que tem o objetivo de fornecer grande poder computacional a médio e longo prazo, disponibilizando os recursos ociosos dos vários nós formadores do Grid. Não visa grande desempenho instantâneo, mas sim uma curva mais tênue, se o desempenho procurado for a médio ou longo prazo.

Condor foi concebido para ser utilizado em Grid's compostos por Cluster's e computadores de vários tipos conectados à Internet (NOWs), sendo chamado de Condor Pool a NOW que executa o Condor. Dentro do Condor Pool, o elemento mais importante é o Matchmaker que é o responsável pela alocação de tarefas para as diversas máquinas ou Cluster's que compõe o Grid. Esta alocação é feita com base nas necessidades para a resolução de uma tarefa, sendo a mesma definida pelo usuário quando da sua submissão ao Grid. Por outro lado, é possível também

definir as restrições de uso de uma máquina quando a mesma é incluída no Pool. Estas restrições permitem que o dono de uma máquina específica defina como ela será utilizada dentro do Pool, sendo que o meio mais comum é a permissão de sua utilização pelo Pool quando existir algum tipo de ociosidade. Caso o proprietário volte a utilizar o equipamento, o Condor em execução é suspenso. Uma máquina pode também ser restrita a um grupo de usuários ou a um usuário específico que podem nunca utilizar os recursos disponibilizados por esta máquina.

No caso específico da interrupção de um processo, que está sendo executado em uma máquina Condor, pela retomada de seu uso pelo proprietário, um mecanismo de checkpoint assegura que a tarefa que foi interrompida seja redistribuída para uma outra máquina formadora do Pool, exatamente a partir do ponto da interrupção.

Através do mecanismo de escalonamento ClassAd, Condor pode encontrar dentro do Pool de máquinas formadoras de sua comunidade, a ou as máquinas que possuam os recursos ociosos e que melhor atendam a uma determinada aplicação. O ClassAd é também responsável por agregar e disponibilizar recursos de novas máquinas quando estas são inseridas no Pool.

#### **4.2.2 CONDOR G**

Condor G significa a união das tecnologias Globus e Condor que visa o compartilhamento de recursos entre várias instituições. Globus fornece os protocolos necessários para a interconexão segura entre os chamados interdomínios bem como uma padronização para sistemas de batch remoto.

Por outro lado, Condor é o responsável pela submissão de tarefas, alocação de recursos, recuperação de erros e criação amigável de ambientes de execução. O resultado é extremamente benéfico para o usuário final porque este pode utilizar uma ampla gama de recursos disponibilizados entre várias instituições.

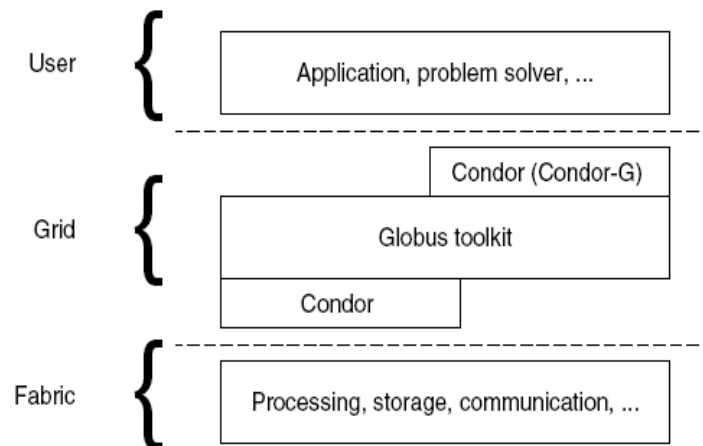


Figura 7 – Condor G Grid middleware

A figura acima mostra de forma bastante clara que o Globus toolkit atua como uma ponte provendo serviços de comunicação entre uma aplicação e uma máquina ou Cluster remoto. A tecnologia Condor é usada como frontend e backend sendo que Condor-G provê recursos de submissão e gerenciamento de serviços.

#### 4.2.3 FERRAMENTAS CONDOR

Condor provê um conjunto de ferramentas utilizadas para o gerenciamento de processos e submissão de tarefas. As principais ferramentas utilizadas são:

GridFTP – Extensão do protocolo FTP para transferência segura de dados. Dentre suas funcionalidades cabe destaque para o; Controle Externo de Transferência, que permite que um usuário inicie, monitore e controle de forma remota a transferência de dados; a Autenticação, Integridade e Confidencialidade, que a partir da utilização do Generic Security Service permite a validação das partes envolvidas na comunicação; a Transferência Particionada dos dados, onde um conjunto de máquinas envia por vários canais suas informações disponíveis; A Negociação Automática do Buffer TCP, onde o tamanho do Buffer pode aumentar ou diminuir, garantindo assim um bom desempenho na comunicação; a Transmissão de Dados Parcial, onde apenas um determinado conjunto de informações é transferido e a Transferência Paralela, onde vários canais TCP são utilizados paralelamente para a transmissão da informação.

Stork – Agendador das operações de transferência de dados no Grid e responsável também pela execução, monitoramento e pela gerência em caso de falhas, permitindo a retomada destas transferências.

Chirp – Protocolo responsável pelo acesso aos dados por uma aplicação Condor. O uso do NeST provê suporte a vários protocolos de armazenamento tais quais NFS e GridFTP.

DiskRouter – Permite a otimização da transferência de dados em redes de longa distância. A partir da utilização de buffers em disco e memória em estações intermediárias na rede consegue melhorar de forma considerável o desempenho.

Batch-Aware Distributed File System – Implementa um sistema de arquivos distribuído com melhores características se comparado a outros sistemas similares distribuído. Faz uso de componentes do Condor, tal qual o NeST, Parrot e ClassAd.

## **5 METODOLOGIA UTILIZADA**

A metodologia utilizada foi a de pesquisa bibliográfica, buscando em várias fontes subsídios necessários para a elaboração deste trabalho. Como a popularização do assunto é relativamente atual, é importante lembrar que o conceito de Grid Computacional existe há algum tempo, todavia, o mesmo era praticamente exclusivo dos ambientes científicos e de pesquisa, só agora se tornando uma tecnologia de abrangência pública. A utilização da Internet como fonte de pesquisa foi amplamente aplicada. Nela é possível encontrar grande variedade de material, desde tutoriais completos sobre como instalar um sistema tipo Globus até short papers e publicações diversas.

## **6 CONCLUSÃO FINAL E POSSÍVEIS TRABALHOS FUTUROS**

Dentre todas as ferramentas utilizadas no gerenciamento de um Grid Computacionais, Globus e Condor tem se destacado devido à integração entre as mesmas. Globus é indicada para host's individuais, fornecendo todos os protocolos necessários para a interconexão, enquanto Condor, foi concebido para ser utilizado em Grid's compostos por Cluster's e computadores de vários tipos conectados à Internet. Através de Condor G as duas tecnologias se completam.

Novas ferramentas de gerenciamento, bem como suítes mais simples de serem instaladas irão surgir com o passar do tempo, juntamente com a maior difusão da tecnologia. Um campo ainda pouco explorado é o da utilização de Grid's associado com a tecnologia Wireless. Alguns trabalhos já existem nesta área e com a diminuição dos custos e a difusão das redes Wireless, certamente é um campo promissor, pois novas formas de gerenciamento se farão necessárias.



## 7 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] <http://www-formal.stanford.edu/jmc> - página pessoal de John MacCarthy. Acessada em 26 de agosto de 2005.
- [2] <http://fp.mcs.anl.gov/~foster> – página pessoal de Ian Foster. Acessada em 26 de agosto de 2006.
- [3] BYRNE, J. A. **The virtual corporation**. Revista Business Week, Fevereiro, 1993.
- [4] JÄGERS, Hans; JANSEN, Wendy; STEENBAKKERS, Wilchard. **Characteristics of Virtual organizations**. In: SIEBER, Pascal; GRIESE, Joachim (eds.). Organizational Virtualness. Proceedings of the VoNet - Workshop, April 27-28, 1998. Bern, Simowa Welag, 1998. p. 65-76.
- [5] MOWSHOWITZ, A. **Virtual organization: a vision of management in the information age**. In: The Information Society. v. 10, United Kingdom, 1994, p. 267 - 288.
- [6] BULTJE, René; VAN WIJK, Jacoliene: **Taxonomy of Virtual Organisations, based on definitions, characteristics and typology.**, in: VoNet: The Newsletter , 2 (1998) 3, p. 16.
- [7] FOSTER I.; KESSELMAN, C.; TUECKE, S. **The Anatomy of the Grid: Enabling Scalable Virtual Organizations** 2001. PDF disponível em <http://www.globus.org/alliance/publications/papers.php>. Acesso em: 27 julho de 2005.
- [8] <http://setiathome.berkeley.edu> - página oficial do projeto de busca de vida inteligente fora da terra. Acessado em 03 de agosto de 2005.
- [9] <http://www.gridforum.org> - página oficial do Global Grid Fórum. Acessado em 28 de julho de 2005
- [10] FOSTER I.; GEISLER J.; NICKELSS B.; et tal **Software Infrastructure for the I-WAY High-Performance Distributed Computing Experiment**. Artigo disponível em <http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/hpdc/&toc=comp/proceedings/hpdc/1996/7582/00/7582toc.xml>. Acessado e adquirido em 05 de setembro de 2005.
- [11] **The Globus Alliance**. Disponível em: <http://www.globus.org>. Acesso 17 de setembro de 2005.
- [12] FOSTER I.; KESSELMAN C. **The Globus Project: A Status Report**, 1998. PDF disponível em <http://www.globus.org/alliance/publications/papers.php>. Acesso em 27 de julho de 2005.
- [13] FOSTER I; KESSELMAN, Carl. **Globus: A Metacomputing Infrastructure Toolkit**. PDF disponível em <http://www.globus.org/alliance/publications/papers.php>. Acesso em: 27 de julho de 2005.

- [14] **Condor High Throughput Computing**. PDF disponível em: <http://www.cs.wisc.edu/condor/>. Acesso em: 02 de setembro de 2005.
- [15] DOUGLAS, Thain; TANNENBAUM, Todd; LIVNY Miron. **Condor and the Grid – Capítulo 11**. PDF disponível em [http://media.wiley.com/product\\_data/excerpt/90/04708531/0470853190.pdf](http://media.wiley.com/product_data/excerpt/90/04708531/0470853190.pdf). Acesso em: 25 de julho de 2005.
- [16] KASSELMAN C.; **The Grid: Blueprint for a New Computing Infrastructure**, 1998.
- [17] FERREIRA, E.C.; SANTOS, A.F.P dos; SCHULZE, B. **Distribuição de Processos em Ambiente de Grid**. Disponível em: <http://arquivosweb.lncc.br/pdfs/relatorio-31-03.pdf>. Acesso em: 04 de setembro de 2005.
- [18] **The DataGrid Project**. PDF disponível em: <http://eu-datagrid.web.cern.ch/eu-datagrid/>. Acesso em: 7 de setembro de 2005.
- [19] FOSTER, I; KESSELMAN, Carl. **Computational Grids**. Capítulo 2 de **The Grid: Blueprint for a New Computing Infrastructure** Capítulo 2, Morgan-Kaufman, 1999. PDF disponível em: <http://www.globus.org/alliance/publications/papers/chapter2.pdf>. Acesso em: 27 de julho de 2005.
- [20] FOSTER, I. **What is the Grid? A Three Point Checklist**. PDF disponível em: <http://www.globus.org/alliance/publications/papers.php>. Acesso em: 27 de julho de 2005.
- [21] FOSTER, I; KESSELMAN, Carl; NICK, J.; et al **The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration**. PDF disponível em: <http://www.globus.org/alliance/publications/papers.php>. Acesso em: 27 de julho de 2005.
- [22] SCHOPF J.M.; NITZBERG, B. **Grids: Top Ten Questions**. PDF disponível em <http://www.globus.org/alliance/publications/papers.php>. Acesso em: 27 de julho de 2005.
- [23] CZAJKOWSKI Karl; FERGUSON, F Donald; FOSTER, Ian; et al **The Ws-Resource Framework** PDF disponível em <http://www.globus.org/wsrf>. Acessado em 20 de novembro de 2005.
- [24] CZAJKOWSKI Karl; FERGUSON, F Donald; FOSTER, Ian; et al **From Open Grid Service Infrastructure to WS-Resource Framework: Refactoring & Evolution**. PDF disponível em <http://www.globus.org/wsrf>. Acessado em 20 de novembro de 2005.
- [25] NETO, W.; CIRNE, E.Santos. **Grids Computacionais: Da Computação de Alto Desempenho a Serviços sob Demanda**. PDF. SBRC 2005.

[26] SRINIVASAN, Latha; **WSRF-WS-ResourceLifetime-1.2-draft-03**. PDF disponível em <http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ResourceLifetime-1.2-draft-03.pdf>. Acessado em 20 de novembro de 2005.

[27] GRAHAM, Steve; TREADWELL, Jem. **WSRF-WS-ResourceProperties-1.2-draft-04**. PDF disponível em <http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ResourceProperties-1.2-draft-04.pdf>. Acessado em 20 de novembro de 2005.

[28] MAGUIRE, Tom; BANKS, Tim; SNELLING, David. **WSRF-Ws\_Service\_Group-1.2-spec-pr-02**. Pdf disponível em [http://docs.oasis-open.org/wsr/wsr-ws\\_service\\_group-1.2-spec-pr-02.pdf](http://docs.oasis-open.org/wsr/wsr-ws_service_group-1.2-spec-pr-02.pdf). Acessado em 20 de novembro de 2005.

[29] SAM, Liu Lily Meder. **WSRF-WS\_Base\_Faults-1.2-spec-pr-02**. PDF disponível em [http://docs.oasis-open.org/wsr/wsr-ws\\_base\\_faults-1.2-spec-pr-02.pdf](http://docs.oasis-open.org/wsr/wsr-ws_base_faults-1.2-spec-pr-02.pdf). Acessado em 20 de novembro de 2005.